WEB APPLICATION DEVELOPMENT USING NODEJS
(Tutorial 2)

Exercise 01

Http server:

```
const http = require("http")

const server = http.createServer((req, res) => {
    res.writeHead(200, {
        'Content-Type': 'text/html charset=utf-8'
    })
    return res.end(`html code`)
})

server.listen(8080, () => {
    console.log("Server is running");
})
```

UML Routing:

```
const URL = require("url");
let url = URL.parse(req.url)
if(url.pathname == '/products') {
    ...
} else if(url.pathname == '/orders') {
    ...
}
```

GET request processing:

```
const queryString = require("querystring")
const URL = require("url")

let url = URL.parse(req.url)
let query = queryString.decode(url.query)
let { product_id, cat_id } = query
```

Exercise 02

NodeJS FS:
https://nodejs.org/api/fs.html

Read file data:

```
const fs = require("fs")
const path = require('path')

let file_path = path.join(__dirname, "views/login.html")
let html = fs.readFileSync(file_path)
return res.end(html)
```

POST request processing:

```
let body = ''
req.on('data', (data) => {
    body = body + data;
})
req.on('end', () => {
    body = queryString.decode(body)
    let { email, password } = body
    ...
})

=============================================
function parseBody(req) {
    return new Promise((resolve, reject) => {
        try {
            let body = '';
            req.on('data', (data) => {
                body = body + data;
            })
            req.on('end', () => {
                body = queryString.decode(body)
                return resolve(body)
            })
        } catch (error) {
            return reject(error)
        }

    })
}
```

Exercise 03

Map & Set:

https://javascript.info/map-set

```
let students = new Map()

students.set(1, {
    id: 1,
    name: "Nguyen Van A"
})
students.set(2, {
    id: 2,
    name: "Nguyen Van B"
})

students.values() => [
    { id: 1, name: "Nguyen Van A"},
    { id: 2, name: "Nguyen Van B"}
]
students.has(1) => true
students.has(2) => true
students.has(3) => false
students.get(1) => { id: 1, name: "Nguyen Van A"}
students.delete(1)
```

JSON response:

```
res.writeHead(200, {
    'Content-Type': 'application/json charset=utf-8'
})


return res.end(JSON.stringify(students))
```

Use Regex:

https://www.tutorialspoint.com/es6/es6_regexp.htm

```
if (url.pathname === '/students') {
    if (req.method === 'GET') {
        ...
    } else if (req.method === 'POST') {
        ...
    }
}
let pattern = /^\/students\/\d+$/g
if (url.pathname.match(pattern)) {
    console.log("matched")
    if (req.method === 'GET') {
        ...
    } else if (req.method === 'PUT') {
        ...
    } else if (req.method === 'DELETE') {
       ...
    }
} else {
    return res.end(JSON.stringify({
        code: 104,
        message: "Invalid url"
    }))
}
```